GATE CS Topic wise Questions
Algorithms

YEAR 2001

Question. 1

Randomized quicksort is an extension of quicksort where the pivot is chosen randomly. What is the worst case complexity of sorting n numbers using randomized quicksort ?

(A) $0(n)$ (B) $0(n \log n)$

(C) $0(n^2)$ (D) $0(n!)$

SOLUTION

In randomized quicksort pivot is chosen randomly, the case complexity of sorting n. In that case the worst case $0(n^2)$ of quicksort become $0(n \log n)$ of randomize quicksort.
Hence (B) is correct option.

Question. 2

Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array $(i \leq n)$, the index of the parent is

(A) $i - 1$ (B) $\left\lfloor \frac{i}{2} \right\rfloor$

(C) $\left\lceil \frac{i}{2} \right\rceil$ (D) $\frac{(i+1)}{2}$

SOLUTION

For an element at index j its children are at $2j$ & $2j+1$ index

let $\quad 2j = i \qquad\qquad 2j+1 = i$

$$j = \frac{i}{2} \qquad\qquad j = \frac{i-1}{2}$$

So $\quad j = \left[\frac{i}{2}\right]$

Hence (B) is correct option.

Question. 3

Let $f(n) = n^2 \log n$ and $g(n) = n(\log n)^{10}$ be two positive functions of n. Which of the following statements is correct ?

(A) $f(n) = 0(g(n))$ and $g(n) \neq 0(f(n))$

(B) $g(n) = 0(f(n))$ and $f(n) \neq 0(g(n))$

(C) $f(n) \neq 0(g(n))$ and $g(n) \neq 0(f(n))$

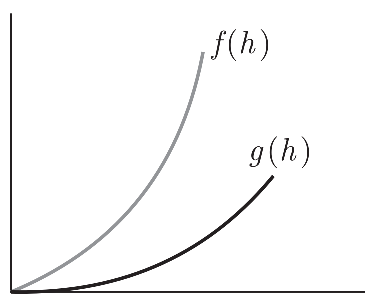
(D) $f(n) = 0(g(n))$ and $g(n) = 0(f(n))$

SOLUTION

$$f(n) = n^2 \log n \ \& \ g(n) = n(\log n)^{10}$$

Since $f(n)$ is polynomially greater than $g(n)$

So $\quad f(n) = 0(g(n))$



But $\quad g(n) \neq 0(f(n))$

Hence (A) is correct option.

Question. 4

Consider the undirected unweighted graph G. Let a breadth-first traversal of G be done starting from a node r. Let $d(r,u)$ and $d(r,v)$

be the lengths of the shortest paths form r to u and v respectively in G. If u is visited before v during the breadth-first traversal, which of the following statements is correct ?

(A) $d(r,u) < d(r,v)$ (B) $d(r,u) > d(r,v)$

(C) $d(r,u) \leq d(r,v)$ (D) None of the above

SOLUTION

In BFS if u is visited before v then either u is some levels before v or u & v are at the same level but u is leftmost in v.

$$d(r,u) \leq d(r,v)$$

Hence (C) is correct option

Question. 5

How many undirected graphs (not necessarily connected) can be constructed out of a given set $V = \{V_1, V_2, ... V_n\}$ of n vertices ?

(A) $\frac{n(n-1)}{2}$ (B) 2^n

(C) $n!$ (D) $2^{\frac{n(n-1)}{2}}$

SOLUTION

Given n vertices the various cases which are possible, no. connection, 1 connection it can be selected ${}^nc_2$ ways, 2 connections & then $(n-1)$ connections to make complete graph.

$$(2)^{{}^nc_2} = 2^{\frac{n(n-1)}{2}}$$

Hence (D) is correct option.

Question. 6

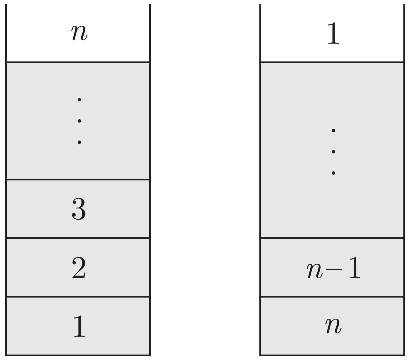
What is the minimum number of stacks of size n required to implement a queue to size n ?

(A) One (B) Two

(C) Three (D) Four

SOLUTION

To implement a queue of size n using stacks each of size n require minimum 2 stacks.



Two stacks to implement queue.

1	2	$\cdots$	n

Hence (B) is correct option.

YEAR 2002

Question. 7

The solution to the recurrence equation $T(2^k) = 3T(2^{k-1}) + 1$, $T(1) = 1$ is

(A) 2^k

(B) $\frac{(2^{k+1} - 1)}{2}$

(C) $3^{\log_2^k}$

(D) $2^{\log_3^k}$

SOLUTION

Given recursion

$$T(2^k) = 3T(2^{k-1}) + 1 \text{ given } T(1) = 1$$

Putting k = 1

$$T(2) = 3T(2^{\circ}) + 1 = 3 \times 1 + 1 = 4$$

$$k = 2 \quad T(4) = 3T(2) + 1 = 3 \times 4 + 1 = 13$$

$$k = 3 \quad T(8) = 3T(4) + 1 = 39 + 1 = 40$$

This sequence of values can be obtained putting $k = 1,2$ & 3. in

$$T(k) = \frac{(3^{k+1} - 1)}{2}$$

Hence (B) is correct option.

Question. 8

The minimum number of colours required to colour the vertices of a cycle with n nodes in such a way that no two adjacent nodes have same colour is.

(A) 2

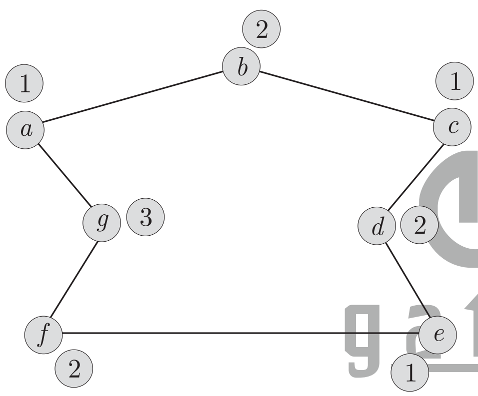
(B) 3

(C) 4

(D) $n-2\left[\frac{n}{2}\right]+2$

SOLUTION

Consider a cycle.



So total no. of colours required to color all vertices of cycle $= 3$

Hence (B) is correct option.

Question. 9

In the worst case, the number of comparisons needed to search a single linked list of length n for a given element is

(A) $\log n$

(B) $\frac{n}{2}$

(C) $\log_2^n - 1$

(D) n

SOLUTION

Worst case of searching occurs when the element to be searched is at the end of the list so no. of comparisons required to search complete list would be n.

Hence (D) is correct option

Question. 10

Maximum number of edges in a n-node undirected graph without self loops is

(A) n^2

(B) $\frac{n(n-1)}{2}$

(C) $(n-1)$

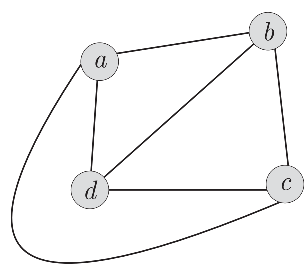
(D) $\frac{(n+1)(n)}{2}$

SOLUTION

Total no. of nodes $= n$

For an edge of n nodes we select any 2 which make a graph.

So $\quad n_{C_2} = \frac{n(n-1)}{2}$ edges



$$n = 4$$
$$\frac{4 \times 3}{2} = 6$$

Hence (B) is correct option.

Question. 11

The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is :

(A) $\frac{n}{2}$

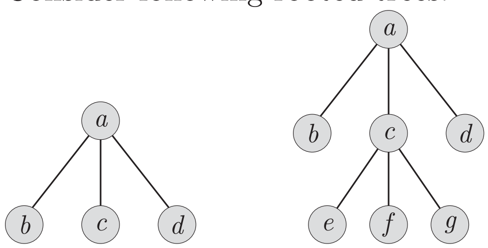
(B) $\frac{(n-1)}{3}$

(C) $\frac{(n-1)}{2}$

(D) $\frac{(2n+1)}{3}$

SOLUTION

Consider following rooted trees.



$$n = 4$$
$$\frac{2n+1}{3} = 3$$

No. of leaf $= 3$

Hence (D) is correct option.

Question. 12

Consider the following algorithm for searching for a given number x in an unsorted array $A[1.....n]$ having n distinct values :

(1) Choose an i uniformly at random from $[1....n]$

(2) If $A[i] = x$ then stop else Goto 1;

Assuming that x is present A, What is the expected number of comparisons made by the algorithm before it terminates ?

(A) n (B) $n-1$

(B) $2n$ (D) $\frac{n}{2}$

SOLUTION

Given array $A[1.....n]$, an element $A[i]$ is chosen randomly from 1 to n.

This would require n selections & comparisons to find x in array.

Hence (A) is correct option.

Question. 13

The running time of the following algorithm

Procedure $A(n)$

If $n <= 2$ return (1) else return $(A(\lceil\sqrt{n}\rceil))$;

Is best described by

(A) $0(n)$ (B) $0(\log n)$

(C) $0(\log\log n)$ (D) $0(1)$

SOLUTION

This is a recursive procedure. Which always calls itself by value $\sqrt{n}$ .So the recursion gas till $n > 2$.

Let $n = 256$

Rec 1 $n = 256$

2 $n = 16$

3 $n = 4$ recursion $= 4$

4 $n = 2$

$\log_2 256 = 8$

Hence (B) is correct option.

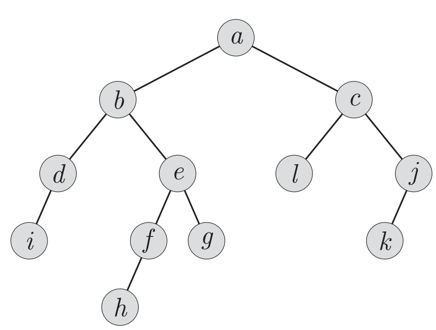
Question. 14

A weight-balanced tree is a binary tree in which for each node, the number of nodes in the let sub tree is at least half and at most twice the number of nodes in the right sub tree. The maximum possible height (number of nodes on the path from the root to the furthest leaf) of such a tree on n nodes is best described by which of the following ?

(A) $\log_2 n$ (B) $\log_{\frac{4}{3}} n$

(C) $\log_3 n$ (D) $\log_{\frac{3}{2}} n$

SOLUTION



$\log_3 12 \cong 4$

Height $= 4$

So $\log_2 n$

In this tree at every node no. of children nodes in left & right subtree follows all properties required & height given by $\log_2 n$.

Hence (A) is correct option.

Question. 15

To evaluate an expression without any embedded function calls

(A) One stack is enough

(B) Two stack are needed

(C) As many stacks as the height of the expression tree are needed

(D) A Turning machine is needed in the general case

SOLUTION

To evaluate an expression we need only 1 stack in which the operands & operators are pushed into, & then evaluated using pop operations.

Hence (A) is correct option.

YEAR 2003

Question. 16

Consider the following three claims

1. $(n+k)^m = \Theta(n^m)$, where k and m are constants
2. $2^{n+1} = O(2^n)$
3. $2^{2^n+1} = O(2^n)$

Which of these claims are correct?

(A) 1 and 2

(B) 1 and 3

(C) 2 and 3

(D) 1, 2 and 3

SOLUTION

(1) $(n+k)^m$ if we expand it.

It is $(n^m + ...)$

During tight bound $\theta(n^m)$ correct

(2) $2^{n+1} = 2.2^n \Rightarrow \Theta(2.2^n)$

$\Rightarrow \Theta(2^n)$ correct

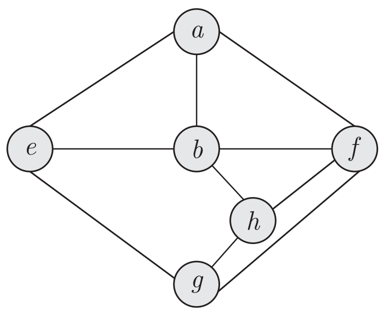
(3) $2^{2n+1} \Rightarrow 2.2^{2^n} \Rightarrow \Theta(2.2^{2^n})$

$\Theta(2^{2^n}) \neq \Theta(2^n)$ incorrect

Hence option (A) is correct

Question. 17

Consider the following graph



Among the following sequences

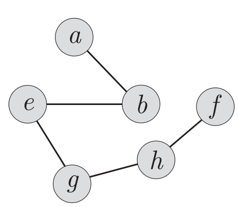
1. *abeghf*
2. *abfehg*
3. *abfhge*
4. *agfhbe*

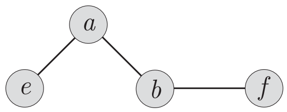
Which are depth first traversals of the above graph?

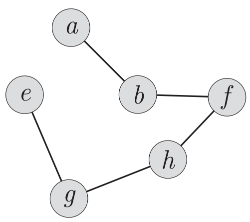
(A) 1, 2 and 4 only
(B) 1 and 4 only
(C) 2, 3 and 4 only
(D) 1, 3 and 4 only

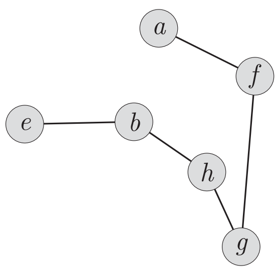
SOLUTION

DFS traversal takes the path to the end & then move to other branch.









Hence option (D) is correct.

Question. 18

The usual $\Theta(n^2)$ implementation of Insertion Sort to sort ab array uses linear search to identify the position where an element is to be inserted into the already sorted part of the array. If, instead, we use binary search to identify the position, the worst case running time will

(A) remain $\Theta(n^2)$ (B) become $\Theta(n(\log n)^2)$

(C) become $\Theta(n\log n)$ (D) become $\Theta(n)$

SOLUTION

Binary search is efficient when the sorted sequence is there, but the worst case scenario for insertion sort would not be sorted sequence so even using binary search instead of linear search the complexity of comparisons will remain $\Theta(n^2)$.

Question. 19

In a heap with n elements with the smallest element at the root, the 7^{th} smallest element ban be found in time

(A) $\Theta(n \log n)$ (B) $\Theta(n)$

(C) $\Theta(\log n)$ (D) $\Theta(1)$

SOLUTION

Here we can follow simple procedure, we can rum heap sort for 7 iterations. In each iteration the top most element is smallest, we note & then replace it with the last element, then we run min heapify algorithm, which brings next smallest element on top. This procedure take $0(\log n)$ time.
We need to run it for 7 times. So tight bound $\Theta(7 \log n) = \Theta(\log n)$
Hence option (C) is correct.

Data for Q. 20 & 21 are given below.

Solve the problems and choose the correct answers.

In a permutation $a_1....a_n$ of n distinct integers, an inversion is a pair (a_1, a_j) such that $i < j$ and $a_i > a_j$.

Question. 20

If all permutation are equally likely, what is the expected number of inversions in a randomly chosen permutation of 1.....n?

(A) $n(n-1)/2$ (B) $n(n-1)/4$

(C) $n(n+1)/4$ (D) $2n[\log_2 n]$

SOLUTION

Let $a_1....a_n$ be 1.......3 here $n = 3$ consider all permutation 123, 132, 231, 213, 312, 321.
Let us consider 312 here the inversions are {(3, 1), (3,2)}
So in a randomly chosen permutation we need to select two no. following inversion property on an average.

no. of inversions $\frac{1}{2}\ {}^nC_2$

$$= \frac{1}{2}\frac{n!}{2!(n-2)!} \Rightarrow \frac{1}{2}\frac{n(n-1)}{2}$$

$$= \frac{n(n-1)}{4}$$

$$\text{If } n = 3 \Rightarrow \frac{3(3-1)}{4} = [1.5] \cong 2$$

Solving this question taking a random example would be much easier.
Hence (B) is correct option.

Question. 21

What would be the worst case time complexity of the insertion Sort algorithm, if the inputs are restricted to permutations of $1....n$ with at most n inversions?

(A) $\Theta(n^2)$ (B) $\Theta(n\log n)$

(C) $\Theta(n^{1.5})$ (D) $\Theta(n)$

SOLUTION

Here at most n inversions are allowed, it means $a_i > a_j$ only for n times.

$52134 \rightarrow \{(5,2),(2,1),(5,1),(5,4),(5,3)\}$

Best case of insertion sort when all are sorted takes 0(n) time.
Worst case when reverse sorted take $0(n^2)$.
So here the solution in between $0(n)$ & $0(n^2)$.
Only n inversions means only n comparison required, each comparison take $\log n$ time.
So time complexity is $0(n\log n)$.

Question. 22

The cube root of a natural number n is defined as the largest natural number m such that $m^3 \leq n$. The complexity of computing the cube root of $n(n$ is represented in binary notation) is

(A) $O(n)$ but not $O(n^{0.5})$

(B) $O(n^{0.5})$ but not $O(\log n)^k)$ for any constant $k > 0$

(C) $O(\log n)^k)$ for some constant $k > 0$, but not $O(\log\log n)^m)$ for any constant $m > 0$

(D) $O(\log\log n)^k)$ for some constant $k > 0.5$, but not $O(\log\log n)^{0.5})$

SOLUTION

n is represented in binary let W suppose using K bit.

To calculate its cube root the time taken is $0(\log n)^k$ but it can't be

$0(\log n)^m$ since $K < m, K > 0, m > 0$.

m is the cube root, since we are doing in binary so we take K.

Hence (C) is correct option.

Question. 23

Let $G = (V, E)$ be an undirected graph with a sub-graph $G_1 = (V_1, E_1)$, Weight are assigned to edges of G as follows

$$w(e) = \begin{cases} 0 \text{ if } e \in E, \\ 1 \text{ otherwise} \end{cases}$$

A single-source shortest path algorithm is executed on the weighted graph (V, E, w) with an arbitrary vertex v_1 of V_1 as the source. Which of the following can always be inferred from the path costs computed?

(A) The number of edges in the shortest paths from v_1 to all vertices of G

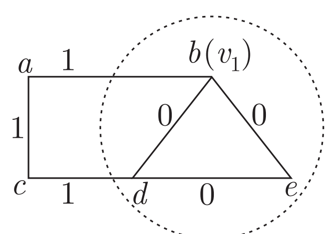
(B) G_1 is connected

(C) V_1 forms a clique in G

(D) G_1 is a tree

SOLUTION

Cligue :- In an undirected graph, a subset of vertices in which every vertex is connected to another vertex by an edge, this subset is called clique.
Consider this graph



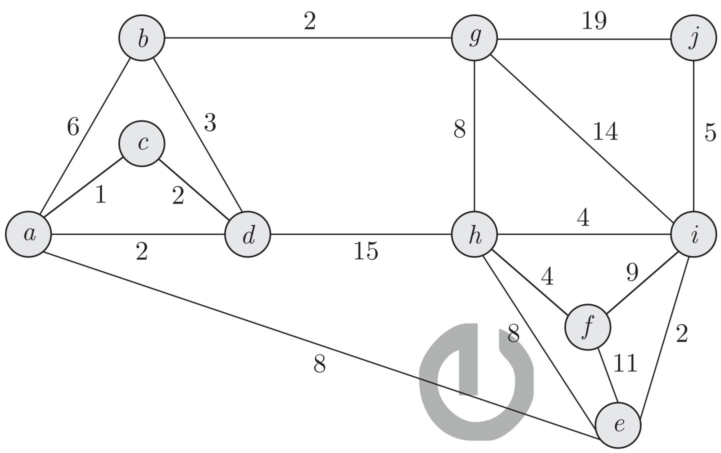
Let b be v_1, Now if we calculate path const.
$v_1 \to a = 1 \quad v_1 \to C = 1 \quad v_1 \to d = 0 \quad v_1 \to e = 0$

(A) So from $v_1 \to C$ we have (b, d) & (c, d) edges i.e. two edges, but the path cost is 1 so (A) false.
(B) Yes G_1 is connected.

(C) If (b, d) not present then G_1 is not a clique.
(D) G_1 can't be a tree since multiple connections between vertices.
Hence (B) is correct option.

Question. 24

What is the weight of a minimum spanning tree of the following graph?



(A) 29
(B) 31
(C) 38
(D) 41

SOLUTION

Starting from vertex a

$\{(a,b),(a,c),(a,d),(a,c)\}$ $\quad$ min $=(a,c)=1$

$\{(a,b),(a,d),(a,e),(c,d)\}$ $\quad$ min $=(a,d)=2$

$\{(a,b),(a,e),(c,d),(b,d),(d,h)\}$ $\quad$ min $=(b,d)=3$

(c,d) not selected since it make cycle

$\{(a,b),(a,e),(d,h),(b,g)\}$ $\quad$ min $=(b,g)=2$

$\{(a,b),(a,e),(d,h),(g,h),(g,j),(g,i)\}$ $\quad$ $(g,h)=8$

$\{(a,e),(d,h),(g,j),(g,i),(h,i),(h,f),(e,h)\}$ $\quad$ $(h,i)=4$

$\{(a,e),(d,h),(g,j),(g,i),(h,f),(e,h)(e,i),(f,i),(j,i)\}$ $\quad$ min $=(e,i)=2$

$\{(a,e),(d,h),(g,j),(g,i),(h,f),(e,h),(f,i),(j,i),(e,f)\}$ $\quad$ $=4$

$\{(a,e),(d,h),(g,j),(g,i),(e,h),(f,i),(j,i),(e,f)\}$ $\quad$ $(j,i)=5$

8 $\quad$ 15 $\quad$ 19 $\quad$ 14 $\quad$ 8 $\quad$ 9 $\quad$ 5 $\quad$ 11

$$Sum\ 1+2+3+2+8+4+2+4+5$$
$$=31$$

Hence (B) is correct option.

Question. 25

The following are the starting and ending times of cetivities A, B, C, D, E, F, G and H respectively in chronological order; "$a_s\, b_s\, a_a\, a_e\, d_s\, a_e\, e_s\, f_s\, b_e\, d_e\, g_s\, e_e\, f_e\, h_s\, g_e\, h_e$'Here, x_s denotes the starting time and x_e denotes the ending time of activity X. W need to schedule the activities in a set of rooms available to us. An activity can be scheduled in a room only if the room is reserved for the activity for its entire duration. What is the minimum number of rooms required?

(A) 3 (B) 4

(C) 5 (D) 6

SOLUTION

Sequence $a_s\, b_s\, c_s\, a_e\, d_s\, c_e\, e_s\, f_s\, b_e\, d_e\, g_s\, e_e\, f_e\, h_s\, g_e\, h_e$
No. of rooms 0 1 2 3 2 3 2 3 4 3 2 3 2 1 2 1 0
Maximum no. of rooms required at a time = 4 option (B).
Here the logic is very simple increase the no. of room if some activity start & decrease by 1 if activity ends.

Question. 26

Let $G = (V, E)$ be a direction graph with n vertices. A path from v_i to v_j in G is sequence of vertices $(v_i, v_{i1}, \ldots.., v_j)$ such that $(v_k, v_{k+1}) \in E$ for all k in i through $j-1$. A simple path is a path in which no vertex appears more than once.

Let A be an $n \times n$ array initialized as follow

$$A[j,k] = \begin{cases} 1 \text{ if (j,k)} \in \text{E} \\ 0 \text{ otherwise} \end{cases}$$

Consider the following algorithm

```
for        o
  for i = 1 to n
     for k = 1 to n
              k =max      k       i     i k
```

Which of the following statements is necessarily true for all j and k after terminal of the above algorithm?

(A) $A[j,k] \leq n$

(B) If $A[j,j] \leq n-1$, then G has a Haniltonian cycle

(C) If there exists a path from j to $k, A[j,k]$ contains the longest path lens from j to k

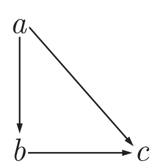
(D) If there exists a path from j to k, every simple path from j to k contain most $A[j,k]$ edges

SOLUTION

Here A during initialization gives the adjacency matrix for directed graph $G(V,E)$. And for very (j,k) we calculate $A[j,k.]$ which stores maximum of the sum of edges making a simple path.

So if there exists a simple path from j to K . $A[j,k]$ contain no of edges in that path.

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{array}{c c} & \begin{array}{ccc} a & b & c \end{array} \\ \begin{array}{c} a \\ b \\ c \end{array} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$



We can see in this example

Hence (D) is correct option.

YEAR 2004

Question. 27

Level order traversal of a rooted tree can be done by stating from the root and performing

(A) preorder traversal
(B) inorder traversal
(C) depth first search
(D) breadth first search

SOLUTION

Level order traversal is done by traversing all the vertices in a particular level & them moving to next level.
This is some as breadth first search where level by level search is done.
Hence (D) is correct option.

Question. 28

Given the following input(4322,1334,1471,9679,1989,6171,6173,4199) and the hash function x mod 10, which of the following statements are true?

1. 9679,1989,4199 hash to the same value
2. 1471,6171 hash to the same value
3. All element hashes to a different value

(A) 1 only
(B) 2 only
(C) 1 and 2 only
(D) 3 and 4 only

SOLUTION

Hash function X mod 10

4322 → 2
1334 → 4
1471 → 1
9679 → 9
1989 → 9
6171 → 1
6173 → 3
4199 → 9

Statement 1 & 2 are correct since. 9679, 1989, 4199, hash to same value.

Question. 29

The tightest lower bound on the number of comparisons, in the worst case, for comparision-based sorting is of the order of

(A) n
(B) n^2
(C) $n \log n$
(D) $n \log^2 n$

SOLUTION

Sorting worst case occurs when arrays are reverse sorted, we need to select every element once & for every element min no. of comparison might be $\log_2 n$.

So overall min. complexity $0(n \log n)$

Hence (C) is correct option.

Question. 30

Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely?

1. preorder and postorderr
2. inorderr and postorder
3. preorder and inorder
4. level order and postorder

(A) 1 only (B) 2 and 3

(C) 3 only (D) 4 only

SOLUTION

For a tree we not only require in order & preorder but also post order traversal.

Preorder & post order help to determine the roots of binary subtrees, inorder arranges those roots in order.

Hence (B) is correct option.

Question. 31

Two matrices M_1 and M_2 are to be stored in arrays A and B respectively. Each array can be stored either in row-major or column-major order in contiguous memory locations. The time complexity of an algorithm to compute $M_1 \times M_2 \times$ will be

(A) best if A is in row-major, and B is in column-major order

(B) best if both are in row-major order

(C) best if both are in column-major order

(D) independent of the storage scheme

SOLUTION

Since the matrices are stored in array, there is no dependence of time complexity on row major or column major. Here only the starting address is known & on the basis of indexes the next memory locations are calculated.
Hence (D) is correct option.

Question. 32

Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?

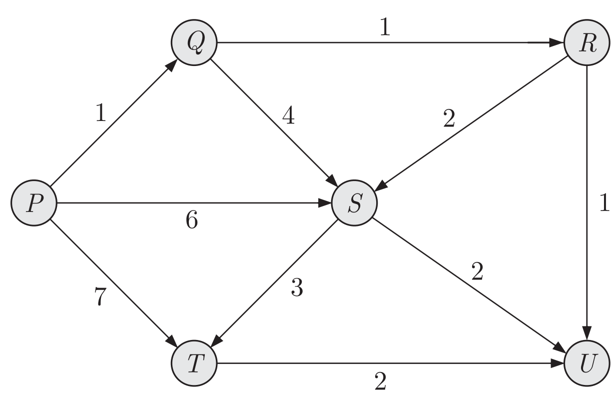
(A) union only
(B) intersection, membership
(C) membership, cardinality
(D) union, intersection

SOLUTION

Membership & cardinality functions takes constt. time i.e. 0(1), but union & intersection require emparison of 1 element with all the other elements so these two would be slowest.
Hence (D) is correct option.

Question. 33

Suppose we run Dijkstra's single source shortest-path algorithm on the following edge-weighted directed graph with vertex P as as the source.

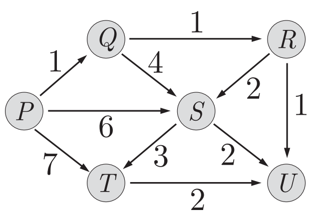


In what order do the nodes get included into the set of vertices ofr which the shortest path distances are finalized?

(A) P, Q, R, S, T, U
(B) P, Q, R, U, S, T

(C) P, Q, R, U, T, S (D) P, Q, T, R, U, S

SOLUTION



	Vertex	I(P)	II(P)	III(Q)	IV(R)	V(V)	VI(S)
I	P	0	0	0	0	0	0
II	Q	∞	$\underline{1}$	1	1	1	1
III	R	∞	∞	$\underline{2}$	2	2	2
IV	S	∞	$\underline{6}$	$\underline{5}$	$\underline{4}$	4	4
V	T	∞	$\underline{7}$	7	7	7	$\underline{7}$
VI	U	∞	∞	∞	$\underline{3}$	3	3

P is source so dist 0, underlined are the adjacent nodes of the current vertex $\angle i$ gives the visited vertex at sequence i.I(P) means P is current vertex.

So order is P Q R U S T

Hence (B) is correct option.

Question. 34

Let $A[1,....n]$ be an array storing a bit (1 or 0) at each location, and $f(m)$ is a function whose time complexity is $\theta(m)$. Consider the

following program fragment written in a C like language:

```
counter   ;
          for       ;    n;
     {if           i      counter++
     else{f(counter); counter   ;}
     }
```

The complexity of this program fragment is

(A) $\Omega(n^2)$ (B) $\Omega(n \log n)$ and $O(n^2)$

(C) $\theta(n)$ (D) $o(n)$

SOLUTION

Here the fragment of code contains for loop which goes from 1 to n.
Since due to given conditions $m < n$.
So complexity of code is $\ominus(n)$
Hence (C) is correct option.

Question. 35

The time complexity of the following C function is (assume $n > 0$)

```
int recursive(int n)     {
     if (n    )
     return(1);
     else
     return(recursive(n  1)+recursive(n  1));
}
```

(A) $O(n)$ (B) $O(n \log n)$
(C) $O(n^2)$ (D) $O(2^n)$

SOLUTION

Given recursion function can be changed into recursive equations i.e.
$T(1) = 1$

$$T(n) = 2T(n-1) \text{ for } n > 1$$

Let $n = 3$

$$\begin{aligned}
T(3) &= T(2) + T(2) \\
T(2) &= T(1) + T(1) \\
&= 1 + 1 = 2 \\
T(3) &= 2 + 2 = 4 \\
T(1) &= 2^0 \\
T(2) &= 2^1 \\
T(3) &= 2^2 \\
T(n) &= 2^{n-1} \\
T(n) &\; 0(2^n)
\end{aligned}$$

Hence (D) is correct option.

Question. 36

The recurrence equation

$$T(1) = 1$$

$$T(n) = 2T(n-1) + n, n \leq 2$$

evaluates to

(A) $2^{n+1} - n - 2$ (B) $2^n - n$

(C) $2^{n+1} - 2n - 2$ (D) $2^n + n$

SOLUTION

Given recurrence.

$$T(n) = 2T(n-1) + n \quad \text{for} \quad n \geq 2$$

Initially $T(1) = 1$

$$T(2) = 2T(1) + 2$$
$$= 2 \cdot 1 + 2 = 4$$
$$T(3) = 2T(2) + 3$$
$$= 2 \cdot 2^2 + 3$$
$$= 2 \cdot 4 + 3 = 11$$
$$T(n-1) = 2T(n-2) + n$$
$$= 2^n - (n-1) - 2$$

So $T(n) = 2^{n+1} - n - 2$

Hence (A) is correct option.

Question. 37

A program takes as input a balanced binary search tree with n leaf modes and computes the value of a function $g(x)$ for each node x. If the cost of computing $g(x)$ is min (number of leaf-nodes in lear-subtree of x, number of leaf-nodes in right-subtree of x) then the worst case time complexity of the program is

(A) $\theta(n)$ (B) $O(n \log n)$

(C) $O(n)^2$ (D) $O(2^n)$

SOLUTION

The function $g(x)$ calculates for a node x min no. of leaf noes whether in left subtree or right subtree. The for balanced BST. The no. of inner nodes = leaf nodes-1

So this loop will sun for max n-times so complexity is $\Theta(n)$

Hence ( ) is correct option.

YEAR 2005

Question. 38

A undirected graph G has n nodes. Its adjacency matrix is given by by an $n \times n$ square matrix whose

1. diagonal elements are $O's$, and
2. non-diagonal elements are 1's. Which one of the following is TRUE?

(A) Graph G has no minimum spanning tree (MST)

(B) Graph G has a unique MST of cost in-1

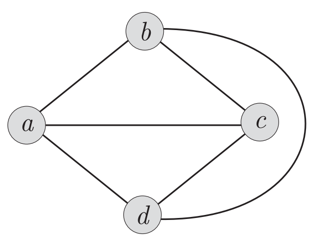
(C) Graph G has multiple distinct MST's, each of cost n-1

(D) Graph G has multiple spanning trees of different costs

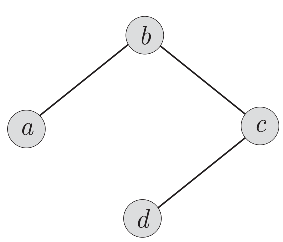
SOLUTION

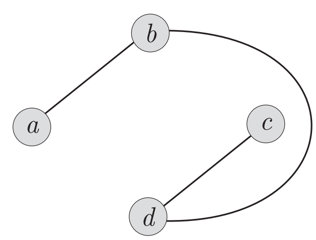
Given adjacency matrix of order 4 is 4×4

0	0	1	1	1
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0



There can be many min spaning but all of $n-1$ cost





So on.
Hence (C) is correct option.

Question. 39

The time complexity of computing the transitive closure of a binary relation on a set of n elements is known to be

(A) $O(n)$

(B) $O(n \log n)$

(C) $O(n^{3/2})$

(D) $O(n^3)$

SOLUTION

Wordshall's algorithm might be used for calculation transitive closure of a set with a elements. This algorithm has complexity $0(n^3)$

In transitive closure two binary relations are there 4 both ranges are the same set.

The require three for loops so $0(n^3)$.

Hence (D) is correct option.

Question. 40

A priority-Queue is implemented as a Max-Heap, Initially, it has 5 elements. The level-order traversal of the heap is given below:

10, 8, 5, 3, 2

Two new elements '1' and '7' are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is

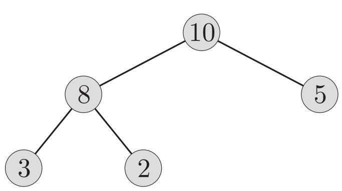
(A) 10, 8, 7, 5, 3, 2, 1

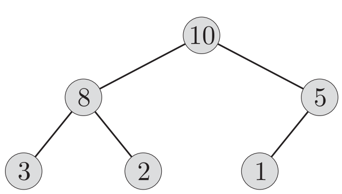
(B) 10, 8, 7, 2, 3, 1, 5

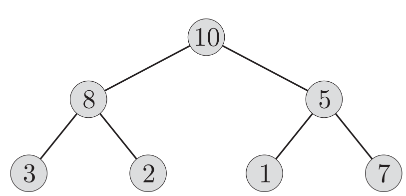
(C) 10, 8, 7, 1, 2, 3, 5

(D) 10, 8, 7, 3, 2, 1, 5

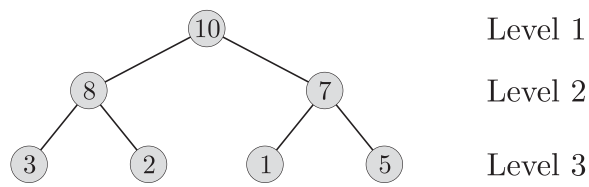
SOLUTION







7 is not in its place/



Level order traversal
10 8 7 3 2 5
Hence (D) is correct option.

Question. 41

How many distinct binary search trees can be created out of 4 distinct keys?

(A) 5 (B) 14

(C) 24 (D) 35

SOLUTION

The formulae for getting no. of different BST for n distinct keys is given by.

$$\frac{1}{n+1}\,{}^{2n}C_n$$

Here $n = 4$

$$= \frac{1}{5}\,{}^{8}C_4$$

$$= \frac{1}{5} \times \frac{8 \times 7 \times 6 \times 5}{4 \times 3 \times 2}$$

$$= 14$$

So 14 distinct BST are possible
Hence (B) is correct option.

Question. 42

In a complete k-ary, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is

(A) $n\ k$ (B) $(n-1)k+1$

(C) $n(k-1)+1$ (D) $n(k-1)$

SOLUTION

No. of internal nodes $= n$
Each node has K children
So total nK

$$\text{Leaf nodes} = nK - n$$
$$= n(K-1)$$

So considering not node also.

$$\text{No. of leaf nodes} = n(K-1)+1$$

Hence (C) is correct option.

Question. 43

Suppose $T(n) = 2T(n/2) + n, T(0) = T(1) = 1$

Which one of the following is FALSE?

(A) $T(n) = O(n^2)$ (B) $T(n) = \theta(n \log n)$
(C) $T(n) = \Omega(n^2)$ (D) $T(n) = O(n \log n)$

SOLUTION

Given recurrence.

$$T(n)\ 2T(n/2) + n$$

Initially $T(0) = T(1) = 1$
Comparing with master's theorem

$$T(n) = aT(n/b) + f(n)$$
$$a = 2\ \ b = 2\ \ \log_b a = 1\ \ f(n) = n$$
$$n^{\log_b^a} = n$$
$$n^{\log_b^a} = f(n)$$

So case II of Master theorem is applied which says.

$$T(n) = \ominus\ (f(n)\log_n \log_b^a)$$
$$= \ominus\ (n \log n)$$

Hence (B) is correct option.

Question. 44

Let $G\ (V, E)$ an undirected graph with positive edge weights. Dijkstra's single source-shortes path algorithm can be implemented using the binary heap data structure with time complexity?

(A) $O(|V|^{2)}$ (B) $O(|E|) + |V| \log |V|)$
(C) $O(|V| \log |V|)$ (D) $O((|E| + |V|) \log |V|)$

SOLUTION

Diykstra Algorithm for every vertex we consider the binary heap to find shortest path. This take V logV time.
And we need to transverse each edge 1 time atleast.
So overall complexity $0(E+V\log V)$
Which is option (B)
Hence (B) is correct option.

Question. 45

Suppose there are log n sorted lists of $n/\log n$ elements each. The time complexity of producing a sorted list of all these elements is: (Hint: Use a heap data structure)

(A) $O(n\log\log n)$ (B) $\theta(n\log n)$

(C) $\Omega(n\log n)$ (D) $\Omega(n^{3/2})$

SOLUTION

There are $\log n$ sorted lists, with $\frac{n}{\log n}$ elements each, total elements n. We need to merge these heap & procedure sorted. Merging take. $\log n$ time & soting takes $0(n\log n)$.
Overall to produce sorted result take $0(n\log\log n)$
Hence (A) is correct option.

Data for Q. 46 & 47 are given below.

Solve the problems and choose the correct answers.
Consider the following C-function:

```
double foo(int n){
     int i;
double sum;
if   (n    ) return 1.0;
     else {
     sum  0 0;
     for (i  0;  n;i  )
     sum += goo (i);
     return sum;
     }}
```

Question. 46

The space complexity of the above function is

(A) $O(1)$ (B) $O(n)$

(C) $O(n!)$ (D) $O(n^n)$

SOLUTION

We need to store largest variables double sum. The function foo() has recursive calls n times.

For every recursion we need 1 sum variable, size of largest double would be $1 \cdot 2 \cdot 3 \cdot ....n = n!$

So complexity $0(n!)$

This factorial is the no. of sub recursions in every recursion

Hence (C) is correct option.

Question. 47

The space complexity of the above function is foo O and store the values of foo (i),$0 <= i < n$, as and when they are computed. With this modification, the time complexity for function foo$O$ is significantly reduced. The space complexity of the modified function would be:

(A) $O(1)$ (B) $O(n)$

(C) $O(n^2)$ (D) $O(n!)$

SOLUTION

Here we store values in foo(i) only when they are completed then in every recursion we require space to store 1 double.

So overall n calls we require $0(n)$

Hence (B) is correct option.

Data for Q. 48 & 49 are given below,

Solve the problems and choose the correct answers.

We are given 9 tasks $T_1, T_2, ... T_9$. The execution of each task requires one unit of time. We can execute one task at a time. T_i has a profit P_i and a deadline d_i profit P_i is earned if the task is completed before the end of the d_1^{th} unit of time.

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

Question. 48

Are all tasks completed in the schedule that gives maximum profit?

(A) All tasks are completed

(B) T_1 and T_6 are left out

(C) T_1 and T_8 are left out

(D) T_4 and T_6 are left out

SOLUTION

Arranging the data in increasing order of deadlines & then profit.

Task	7	2	9	4	5	3	6	8	1
Profi15t	23	20	25	18	18	30	10	16	15
Deadtime	2	2	3	3	4	5	5	7	7

Time	Task selected	Profit
1	7	23
2	2	20
3	9	25
4	5	18
5	3	30
6	8	16
7	1	15
SUM		147

So here T_4 & T_6 not selected.

Hence (D) is correct option.

Question. 49

What is the maximum profit earned?

(A) 147

(B) 165

(C) 167

(D) 175

SOLUTION

Solving from the some algorithm solved in previous question the sum

is 147 for the profit.
Hence (A) is correct option.

YEAR 2006

Question. 50

Consider the polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + a_2 x^3$, where $a_i \neq 0, \forall i$. The minimum number of multiplications needed to evaluate p on an input x is

(A) 3
(B) 4
(C) 6
(D) 9

SOLUTION

$$P(x)\ a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

We can reduce the no. of multiplications using Horner's rule.

$$P(x) = a_0 + (a_1 + (a_3 x + a_2) x) x$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ (1) & (2) & (3) \end{array}$$

So min $= 3$
Hence (A) is correct option.

Question. 51

In a binary max heap containing n numbers, the smallest element can be found in time

(A) $\theta(n)$
(B) $\theta(\log n)$
(C) $\theta(\log\log n)$
(D) $\theta(1)$

SOLUTION

Heap is implemented using array & to find maximum or minimum element in array we take only n maximum comparison.
So complexity is $\Theta(n)$
Hence (A) is correct option.

Question. 52

Consider a weighted complete graph G on the vertex set $\{v_1, v_2, ... v_n\}$

such that the weight of the edge (v_i, v_j) is $2\,|\,i-j\,|$. The weight of a minimum spanning tree of G is

(A) $n-1$ (B) $2n-2$

(C) $\binom{n}{2}$ (D)

SOLUTION

Given are n vertices $\{0_1......0_n\}$
A simple min. spanning tree would be like joining edges between only v_i & v_{i-1}.
The weight of every such edge would be

$$\begin{aligned} 2\left| v_i - v_j \right| &= 2\,(i-j) \\ &= 2\,(-\,i+1+i) \\ &= 2 \end{aligned}$$

In complete graph the no. of edges in MST $= n-1$

$$\begin{aligned} \text{So total weight of MST} &= 2 \times (n-1) \\ &= 2n-2 \end{aligned}$$

Hence (B) is correct option.

Question. 53

To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, then data structure to be used is

(A) Queue (B) Stack

(C) Heap (D) B-Tree

SOLUTION

The Best data structure would be heap or priority queue implemented as heap. Here we require the shortest path and in priority queue implementation the priorities may be assigned on the basis of shortest distance, so selection of max priority or min distance takes $0\,(n)$ time.
Hence (C) is correct option.

Question. 54

A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0. The roots is stored at $X[1]$. For a node stored at $X[1]$, the left child, if any, is stored in $X[2\text{i}]$ and the right child, if any, in $X[2i+1]$. To be able to store any binary tree on n vertices, the minimum size of X should be

(A) $\log_2 n$ (B) n
(C) $2n+1$ (D) $2n$

SOLUTION

Right child & left child of element $X[i]$ are shared in array at $X[2i+1]$ & $X[2i]$ respectively & index is at $X[1]$ & $X[2]$ & $X[3]$ are its child. So till index 3 we stored 3 elements & so on.
So we require the array of size n to store n elements.
Hence (B) is correct option.

Question. 55

Which one the following in place sorting algorithms needs the minimum number of swaps?

(A) Quick-sort (B) Insertion sort
(C) Selection sort (D) Heap sort

SOLUTION

Quicksort, selection sort require, at most n swaps per iterations, the same case may occur making heap & keeping max-min property. But in insertion sort, one element is compared to all its pre index elements, but it is swapped only with 1 element but then all the elements need to move 1 index further so no. of swaps increases.
So overall seeing Xeap sort would have min no. of swaps.
Hence (D) is correct option.

Question. 56

Consider the following C-program fragment in which $i, j,$ and n are integer variables.

$$\text{for } (i = n, j = 0; i > 0; i/2, j += i);$$

Let Val (j) =denote the value stored in the variable j after termination of the for loop. Which one of the following is true?

(A) $\text{val}(j) = \theta(\log n)$ (B) $val(j) = \theta(\sqrt{n})$
(C) $\text{val}(j) = \theta(n)$ (D) $\text{val}(j) = \theta(n \log n)$

SOLUTION

Here after every iteration the value of $i = i/2$, & j is the summation of these i till i reaches to 1.

$$j = n + n/2 + n/2^2 ....n/2^{\log_2^n}$$

Sum of this series.

Would give $\ominus(n)$ order.

Hence (C) is correct option.

Question. 57

An element in an array X is called a leader if it is grater than all elements to the right of it in X. The best algorithm to find all leaders in an array.

(A) Solves it in linear time using a left to right pass of the array

(B) Solves in linear time using a right to left pass

(C) Solves it is using divide and conquer in time $\theta(n \log n)$

(D) Solves it in time $\theta(n^2)$

SOLUTION

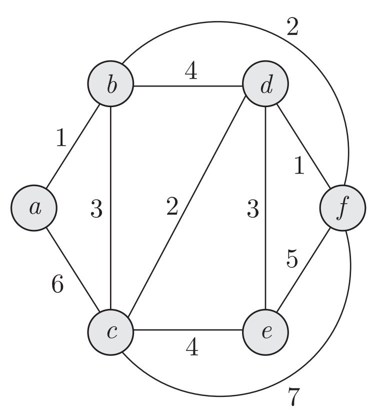
In quick sort (divide & conquer) algorithm after every run we being 1 element at its right place i.e. all the elements in the left are smaller & in the right are greater than it.

So we can apply quick sorts divide & conquer method of complexity $0(n \log n)$ to do this, to check whether all elements in right are smaller than it or not.

Hence (C) is correct option.

Question. 58

Consider the following graph:



Which one of the following cannot be the sequence of edges added, in that order, to a minimum spanning tree using Kruskal's algorithm?

(A) $(a-b),(d-f),(b-f).(d-c),(d-e)$

(B) $(a-b),(d-f),(b-c),(b-f),(d-e)$

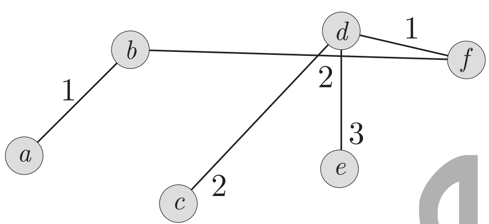
(C) $(d-f),(a-b),(d-c),(d-e),(d-e)$

(D) $(d-f),(a-b),(b-f),(d-e),(d-e)$

SOLUTION

Let us check each option

(A) (a-b), (b-f), (b-f), (d-c), (d-e)



(B) (a-b), (d-f), (d-c), (d-f), (d-e), (d,c) & (b,f) has same weight so correct.

(C) (d-f), (a-b), (d-c), (d-f), (d-e) possible

(D) (d-f), (a-b), (b-f), (d-e) (d-c).

(b-f) has weight = 2

(d-e) has weight = 3

(d-c) has weight = 2

So (d-e) can't be taken before (d-c)

So (d) is incorrect.

Hence (D) is correct option.

Question. 59

Let T be a depth first search tree in a undirected graph G Vertices u and v are leaves of this tree T. The degrees of both u and v in G are at least 2. Which one of the following statements is true?

(A) There must exist a vertex w adjacent to both u and v in G

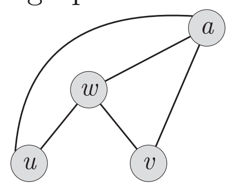
(B) There must exist a vertex w whose removal disconnects u and v in G

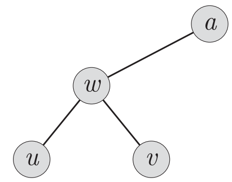
(C) There must be exist a cycle in G containing u and v

(D) There must exist a cycle in G containing u and all its neighbours in G

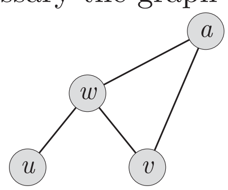
SOLUTION

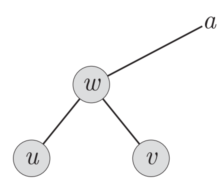
Let the graph G be.





(A) is correct since W is the common vertex.
(B) W is removed but u & v are not dis-connected.
(C) No cycle containing u & v exist.
(D) Not necessary the graph can be also.





Hence (A) is correct option.

Question. 60

A set X can be represented by an array $x[n]$ as follows

$$x[i] = \begin{cases} 1 & \text{if} \quad i \in X \\ 0 & \text{otherwise} \end{cases}$$

Consider the following algorithm in which x, y and z are boolean arrays of size n;

algorithm $zzz(x[], y[], z[])\{$

int i;

for $(i = 0; i < n; ++i)$

$$z[i] = (x[i] \wedge \sim y[i]) \vee (\sim x[i] \wedge y[i])$$

$\}$

The set Z computed by the algorithm is

(A) $(X \cup Y)$
(B) $(X \cap Y)$
(C) $(X - Y) \cap (Y - X)$
(D) $(X - Y) \cup (Y - X)$

SOLUTION

$$\text{Here } z = (x \wedge y') \vee (x' \wedge y)$$

$$(x \cap y') \cup (x' \cap y)$$

This can be written as.

$$(x - y) \cup (y - x)$$

Hence (D) is correct option.

Question. 61

Consider the following is true?

$$T(n) = 2T([\sqrt{n}]) + 1, T(1) = 1$$

Which one of the following is true?

(A) $T(n) = \theta(\log\log n)$ (B) $T(n) = \theta(\log n)$

(C) $T(n) = \theta(\sqrt{n})$ (D) $T(n) = \theta(n)$

SOLUTION

Given recurrence

$$T(n) = 2T([\sqrt{n}]) + 1 \qquad (1)$$

Initially $T(1) = 1$

Solving it by method of change of variables

$$\text{Let } m = \log_2^n \qquad (2)$$

So $n = 2^m$

Putting in equation (1)

$$T(2^m) = 2T(2^{m/2}) + 1$$

This can be re-written as.

$$\mathrm{S(m)} = 2T(m/2) + 1$$

Solving this by Master's method

$$\text{Comparing } T(n) = aT(n/b) + f(n)$$

$$\text{So } n^{\log_b^a} = n^{\log_2^2} = n^\circ = 1 = f(n)$$

So case (2) applies.

$$5(m) = \log m$$

Putting m from equation (2)

$$T(n) = \log\log n$$

$$T(n) = 0(\log\log n)$$

Hence (A) is correct option.

Question. 62

The median of n elements can be found in $O(n)$ time. Which one of the following is correct about the complexity of quick sort, in which remains is selected as pivot?

(A) $\theta(n)$ (B) $\theta(n\log n)$
(C) $\theta(n^2)$ (D) $\theta(n^3)$

SOLUTION

In quick sort the piuot is found in $\log n$ time & this runs for n times. So complexity of Quick sort is $0(n\log n)$ but since given the median as piuot found in $0(n)$
So for n elements to sort this algorithm will take $0(n^2)$
Hence (C) is correct option.

Question. 63

Given two arrays of numbers $a_1.......a_n$ and $b_1,.....b_n$ where each number is 0 or 1, the fastest algorithm to find the largest span (i,j) such that $a_{i+} + a_{i+1} + ... + a_j = b_i + b_{i+1} + ... + b_j$, or report that there is no such span,

(A) Takes $O(3^n)$ and $\Omega(2^n)$ time if hashing is permitted
(B) Takes $O(n^3)$ and $W(n^{2.5})$ time in the key comparison model
(C) Takes $\Theta(n)$ time and space
(D) Takes $O(\sqrt{n})$ time only if the sum of the $2n$ elements is an even number

SOLUTION

Here we require to store two arrays each having n elements.
$0(2n)$ i.e. $\theta(n)$ space complexity. The calculation $a_i + ... + a_j = b_i + ... + b_j$ is to be done n time each such calculation take constant time.
So $0(n)$
Hence (C) is correct.

Question. 64

Consider the following code written in a pass-by reference language like FORTAN and these statements about the code.

```
Subroutine swap i  i
          it  i
L1 :      i   i
L2 :      i   it
     end
```

```
call swap   a
print*,ia,i
end
```

S1: The complier will generate code to allocate a temporary nameless cell, initialize it to 13, and pass the address of the cell to swap

S2: On execution the code will generate a runtime error on line 1.1

S3: On execution the code will generate a runtime error on line 1.2

S4: The program will print 13 and 8

S5: The program will print 13 and-2

Exactly the following set of statement (s) is correct:

(A) $S1$ and$S2$

(B) $S1$ and S4

(C) $S3$

(D) $S1$ amd S5

SOLUTION

S1 : Yes the compiler will generate a temporary nameless cell & initialize it to 13 and pass to swap.
S2 : No error
S3 : No error
S4 : Program will print 13 and 8
S5 : False.
Hence (B) is correct option.

YEAR 2007

Question. 65

The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes is a binary tree of height h is

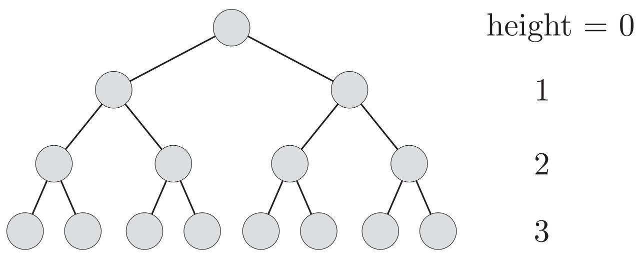
(A) 2^h

(B) $2^{h-1}-1$

(C)$2^{h+1}-1$

(D) 2^{h+1}

SOLUTION



Tree of height	0	1	$2^2 - 1$
	1	3	$2^3 - 1$
	3	15	$2^{h+1} - 1$

Hence (C) is correct option.

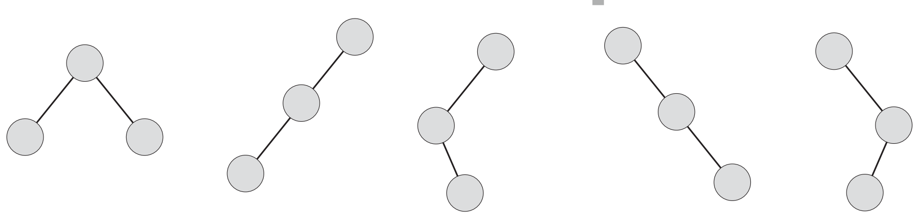
Question. 66

The maximum number of binary trees that can be formed with three unlabeled nodes is

(A) 1 (B) 5

(C) 4 (D) 3

SOLUTION



Mathematically

$$\text{No of binary trees} = \frac{1}{n+1}\ {}^{2n}C_n$$

Here $n = 3$

$$= \frac{1}{4} \times {}^{6}C_3$$

$$= \frac{1}{4} \times \frac{6 \times 5 \times 4}{3 \times 2} = 5 \text{ trees}$$

Hence (B) is correct option.

Question. 67

Which of the following sorting algorithms has the lowest worst-case complexity?

(A) Merge sort (B) Bubble sort

(C) Quick sort (D) Selection sort

SOLUTION

The complexities of worst case when all the elements are reverse sorted for all algorithms are.

Norge $0(n\log_2 n)$
Quick $0(n^2)$
Selection $0(n^2)$
Bubble $0(n^2)$

Merge no has no effect of input nature since it keeps on dividing into 2 problems of size 4/2 so complexity is lower then other three.

Hence (A) is correct option.

Question. 68

The inorder and preorder traversal of a binary tree are

dbeafcg and*abdecfg* respectively

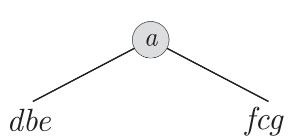
The postorder traversal of the binary tree is

(A) *debfgca* (B) *edbgfca*

(C) *edbfgca* (D) *defgbca*

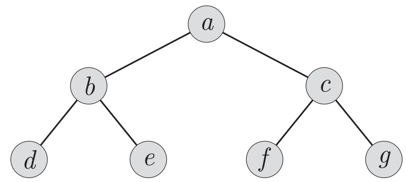
SOLUTION

In order $d\ b\ e\ a\ f\ c\ g$
preorder $a\ b\ d\ e\ c\ f\ g$
1st element of pre order is root



from inorder

in preorder b is before $d\ e$. & c is before $f\ g$.



d e b f g c a

Hence (A) is correct option.

Question. 69

Consider the hash table of size seven, with starting index zero, and a has function $(3x + 4)$ and 7. Assuming the has table is initially empty, which of the following is the contents of the table when the sequence 1,3,8,10 is inserted into the table using closed hashing?

Note that-denotes an empty location in the table

(A) 8,-,-,-,-,-,10
(B) 1,8,10,-,-,-,3
(C)1,-,-,-,-,-,,3
(D) 1,10,8,-,-,-,3

SOLUTION

Hash table

0	1
1	8
2	10
3	-
4	-
5	-
6	3

Inputs.

X	3X+4	mod 7
1	7	mod 7
3	13	mod 7
8	28	mod 7
10	34	mod 7

Hence (B) is correct option.

Question. 70

In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of time complexity, by

(A) Dijkstra's algorithm starting from S.

(B) Warshall's algorithm

(C) performing a DFS starting from S

(D) preforming a BFS starting from S

SOLUTION

Since the graph is unweighted and undirected so no sense in using Diykstra or Warshall also their complexities are $0(n^2)$ & $0(n^3)$ respectively.
BFS starting from S, traverses all the adjacent nodes, & then their adjacent nodes, this calculates shortest path with min complexity.
Hence (D) is correct option.

Question. 71

A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If $L = 41$, and $I = 10$, what is the value of n

(A) 3 (B) 4

(C) 5 (D) 6

SOLUTION

Each internal node has n children & so total nodes $I \times n$
No. of leaf in them

$$I \times n - 1$$
$$I(n-1)$$

But root can't produce leaf

$$I(n-1) + 1 = L$$
$$n = \frac{L-1}{I} + 1$$
$$n \ \frac{41-1}{10} + 1$$
$$n = 5$$

Hence (C) is correct option.

Question. 72

In the following C function, let $n \leq m$.

```
int gcd  n
{
```

```
    if          return m;
    n   n m;
    return gcd m  n  ;
}
```

How many recursive calls are made by this function?

(A) $\Theta(\log_2 n)$
(B) $\Omega(n)$
(C) $\Theta(\log_2 \log_2 n)$
(D) $\Theta(\sqrt{n})$

SOLUTION

In gcd n is replaced by n/m in every iteration so running time has to be less than $0(n)$ or $\Omega(n)$ even less then $\theta(\sqrt{n})$.
It has to be $\theta(\log_2 n)$ since recursion cause the problem size reduced by $n/2$ every iteration.
Hence (A) is correct option.

Question. 73

What is the time complexity of the following recursive function :

```
    intDoSomething(int n)    {
    if        n
        return 1;
    else
    return(DoSomething(floor(sqrt(n))
    )+   n);
        }
```

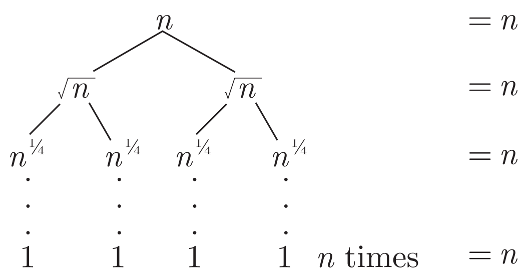
(A) $\Theta(n^2)$
(B) $\Theta(n \log_2 n)$
(C) $\Theta(\log_2 n)$
(D) $\Theta(\log_2 \log_2 n)$

SOLUTION

Here $T(n) = T(\sqrt{n}) + n$

eg $n = 16$ $n = 4$ $n = 2$ so

Recursion tree



At any level K the problem size is $n^{2^{-k}}$ eve keep recursion till this value reach 1.

So
$$\begin{aligned} 2^{-k}\log_2^n &= 1 \\ 2^k &= \log_2 n \\ K\log_2 2 &= \log_2\log_2 n \\ K &= \log_2\log_2 n \\ T(n) &= \theta(\log_2\log_2 n) \end{aligned}$$

Hence (D) is correct option.

Question. 74

Consider the process of inserting an element into a Max Heap, where the Max Heap is represented by an array. Suppose we perform a binary search on the path from the new leaf to the root to find the position for the newly inserted element, the number of comparisons performed is

(A) $\theta(\log_2 n)$ (B) $\theta(\log_2\log_2 n)$

(C) θn (D) $\theta(n\log_2 n)$

SOLUTION

In a Max heap we insert 1 element this takes 0 (1) time since it is an array. Now to find correct position we perform Binary search, & we know BS an array takes $0(\log_2 n)$ time

So (A) is correct option.

Question. 75

Let w be the minimum weight among all edge weights in an undirected connected graph, Let e be a specific edge of weight w. Which of the following is FALSE ?

(A) There is a minimum spanning tree containing e.

(B) If e is not in a minimum spanning tree T,then in the cycle formed by adding e to T, all edges have the same weight.

(C) Every minimum spanning tree has an edge of weight w

(D) e is present in every minimum spanning tree

SOLUTION

Consider every optim separately
(A) since e has min weight w so there would be at test in spanning tree with e
(B) e might not present in MST, but only possible if the other edge taken has also same weight

(C) Every MST must have an edge with w
(D) e might not present in all MST as shown in above example.
Hence (D) is correct option.

Question. 76

An array of n numbers is given, where n is an even number. The maximum as well as the minimum of these n numbers needs to be determined. Which of the following is TRUE about the number of comparisons needed?

(A) At least $2n$-c comparisons, for some constant c, are needed,

(B) At most $1.5n$-2 comparisons are needed.

(C) At least $n, \log_2 n$ comparisons are needed.

(D) None of the above

SOLUTION

One possible way to do this is we select first element as max & also min. Then we compare it with all others. At most this would take 2n comparisons during linear search.
But if we use divide & conquer as for merge sort we have.

$$T(n)\ 2T(n/2) + 2 \text{ for } n > 2$$

Its solution

$$n^{\log_b a} = n^{\circ} = 1 \ \ f(n) > n^{\log_b a} \text{ so case 3.}$$

$$T(n) = \frac{3n}{2} - 2$$

$$= 1.5n - 2$$

Hence (B) is correct option.

Question. 77

Consider the following C code segment:

```
int        IsPrime(n)
           {
                int i,n;
                for (i   ;i   sqrt(n);i   )
                if (n i     )
                print(Not Prime\n  ;
                          return0;)
                return 1;
           }
```

Let $T(n)$ denote the number of times the for loop is executed by the program on input n. Which of the following is TRUE?

(A) $T(n) = O(\sqrt{n})$ and $T(n) + \Omega(\sqrt{n})$

(B) $T(n) = O(\sqrt{n})$ and $T(n) + \Omega(1)$

(C) $T(n) = O(n)$ and $T(n) = \Omega(\sqrt{n})$

(D) None of these

SOLUTION

The loop runs from 2 to $\sqrt{n}$. So maximum iterations can be $\sqrt{n}$. When $n = 2$ loop has only 1 iteration so. 1 & $\sqrt{n}$ are lower & upper bounds respectively.

$$\Omega(1) \leq T(n) \leq 0(n)$$

Hence (B) is correct option.

Data for Q. 78 & 79 are given below.

Solve the problems and choose the correct answers.

Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ respectively.

Question. 78

Which of the following is the Hoffman code for the letters a, b, c, d, e, f ?

(A) 0, 10, 110, 1110, 11110, 11111

(B) 11, 10, 011, 010, 001, 000

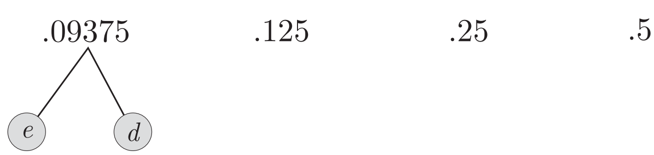
(C) 11, 10, 01, 001, 0001, 0000

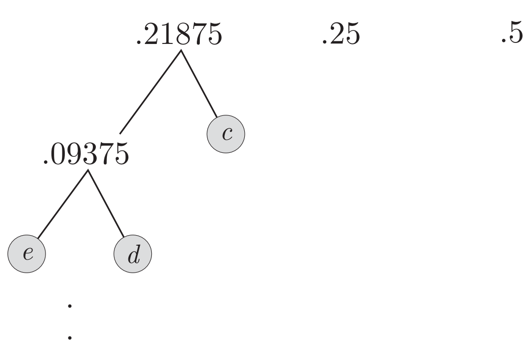
(D) 110, 100, 010, 000, 111

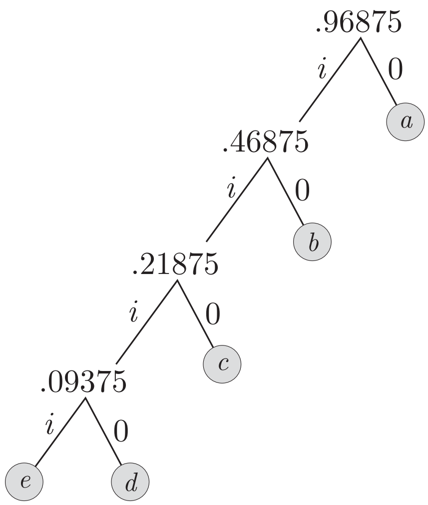
SOLUTION

a	b	d	c	e
.25	.25	.125	.0625	.01325

Arranging in ascending order.







$a = 0$

$b = 0$

$c = 110$

$d = 1110$

$e = 1111$

$$\begin{aligned} a &= 0 \\ b &= 10 \\ c &= 110 \\ d &= 1110 \\ e &= 1111 \end{aligned}$$

Hence (A) is correct option.

Question. 79

What is the average length of the correct answer to Q. ?

(A) 3 (B) 2.1875

(C) 2.25 (D) 1.781

SOLUTION

Length of Huffman code

Length	Prob.	Product
$a \to 1$	1/2	1/2
$b \to 2$	1/4	1/2
$c \to 3$	1/8	3/8
$d \to 4$	1/16	1/4
$e \to 4$	1/32	1/8

Average length $= 1/2 + 1/2 + 3/8 + 1/4 + 1/8$

$$= \frac{4+4+3+2+1}{8}$$

$$= \frac{14}{8}$$

$$= 1.781$$

Hence (D) is correct option.

YEAR 2008

Question. 80

The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity.

(A) $\Theta(n)$ (B) $\Theta(m)$

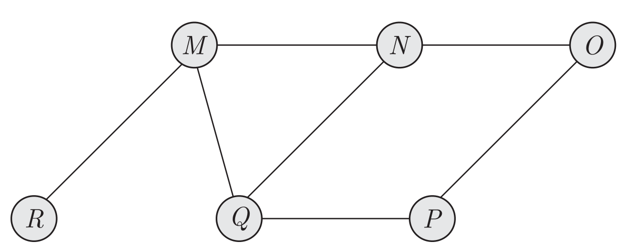
(C) $\Theta(m+n)$ (D) $\Theta(mn)$

SOLUTION

The algorithm we use for finding the number of connected components in an undirected graph on n vertices is to calculate articulation point detection algorithm.
This articulation point divides the graph into 2 connected components.
Complexity of this algorithm is same as 1 DFS run $0(m+n)$ since DFS is the basis of articulation point.
Hence (C) is correct option.

Question. 81

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is



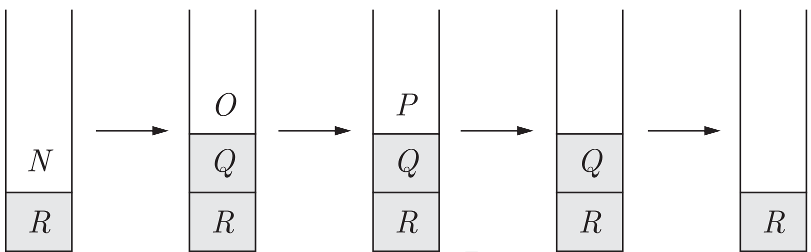
(A) *MNOPQR* (B) *NQMPOR*

(C) *QMNPRO* (D) *QMNPOR*

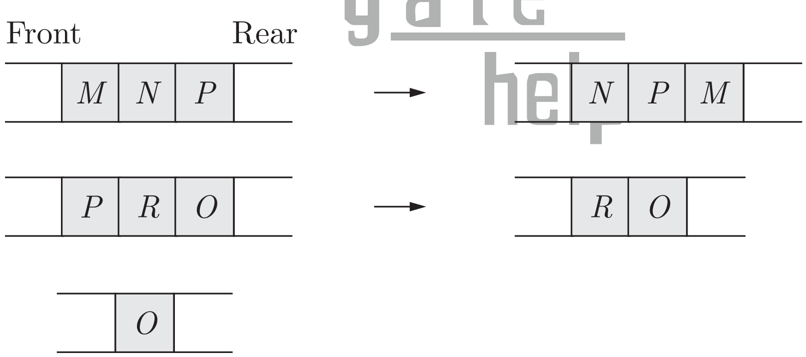
SOLUTION

BFS : Here for every node we visit all its neighbours & then the neighbours of its neighbours.
We use queue to find this.

Start from M→ $M \to N \to O \to P \to Q \to R$



Start from Q $\quad Q \longrightarrow M \longrightarrow N \longrightarrow P \longrightarrow R \longrightarrow O$



Hence (C) is correct option.

Question. 82

Consider the following function;

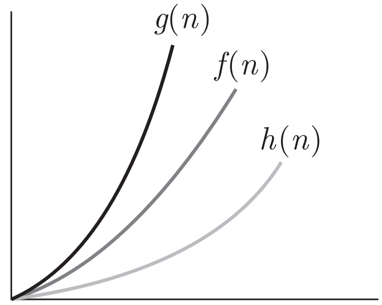
$$f(n) = 2^n$$
$$g(n) = n!$$
$$h(n) = n^{\log n}$$

Which of the following statements about the asymptotic behavior of $f(n)$. $g(n)$ and $h(n)$ is true?

(A) $f(n) = O(g(n)); g(n) = O(h(n))$
(B) $f(n) = \Omega(g(n)); g(n) = O(h(n))$
(C) $g(n) = O(f(n)); h(n) = O(f(n))$
(D) $h(n) = O(f(n)); g(n) = \Omega(f(n))$

SOLUTION

The asymptotic order. $1 < \log\log n < \log n < n^e < n^c < n^{\log n} < c^n < n^n$
$< c^{c^n} < n!$
$f(n) = 2^n \;\; g(n) = n^n \;\; h(n) = n^{\log n}$ from order $h(n) < f(n) < g(n)$



$g(n)$ is the upper bound of $f(n)$
$$g(n) = 0(f(n))$$
$h(n)$ is the lower bound of $f(n)$.
$$h(n) = \Omega(f(n))$$
Hence (D) is correct option.

Question. 83

The minimum number of comparison required to determine if an integer appears more than $n/2$ times in a sorted array of n integers is

(A) $\Theta(n)$
(B) $\Theta(\log n)$
(C) $\Theta(\log * n)$
(D) $\Theta(1)$

SOLUTION

Since all the elements are sorted so we can apply binary search here efficiently. In BS the size of array required to compare reduces by $n/2$ in every iteration.
Here since the sequence is sorted so the same element would come consecutively

$$\begin{aligned} \text{eg } n &= 10 \\ &= 1\ 2\ 2\ 2\ 2\ 2\ 4\ 6\ 8 \\ &\qquad\qquad \uparrow \\ &\qquad\qquad = 2 \text{ correct} \\ &\qquad \uparrow \\ &\qquad = 2 \text{ correct} \\ &\uparrow \\ &= 2 \text{ correct} \\ \lfloor \log_2 10 \rfloor &= 3 \\ &\Theta(\log n) \end{aligned}$$

Hence (B) is correct option.

Question. 84

AB-tree of order 4 is built from scratch by 10 successive insertions. What is the maximum number of node splitting operations that may take place?

(A) 3 (B) 4

(C) 5 (D) 6

SOLUTION

In B tree the data is stored at leaves only a particular node can have maximum. 3 keys, so when 4^{th} insertion comes first split is required, during 7^{th} second split & so on, so for 10 insertions max. 3 splits are required.

We can prove it mathematically.

$$\text{No. of split} \leq 1 + \log_{m/2}\left\lceil \frac{n}{b} \right\rceil$$

$$\text{Here } m \text{ order} = 4 \;\; n = 10 \;\; b = 3$$

$$\begin{aligned} K &\leq 1 + \log_2 \left\lceil \frac{10}{3} \right\rceil \Rightarrow 1 + \log_2^4 \\ K &\leq B \end{aligned}$$

Hence (A) is correct option.

Question. 85

G is a graph on n vertices and $2n-2$ edges. The edges of G can be partitioned into two edge-disjiont spanning trees. Which of the following is NOT true for G?

(A) For every subset of k vertices, the induced sub graph has a most

$2k-2$ edges.

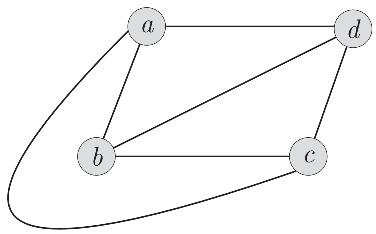
(B) The minimum cut in G has a least two edges

(C) There are two edges-disjoint paths between every pair of vertices

(D) There are two vertex-disjoint paths between every pair of vertices.

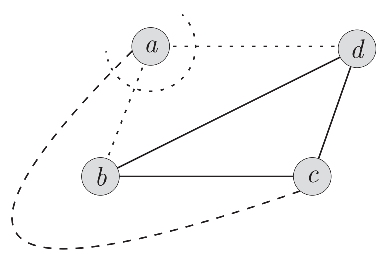
SOLUTION

Consider this graph with $n=4$



Two spanning trees

Statement (B), (C) & (D) are correct.



min cent has 3 edges.

Two edge & vertex disjoint paths are present can be seen in two spanning trees but option (A) is false for $K=2$ $2K-2=i.e$ 2 edges should be there but it is not true.

Question. 86

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into sub-lists each of which contains at least one-fifth of the elements. Let $T(n)$ be the number of comparisons required to sort n elements. Then

(A) $T(n) \leq 2T(n/5) + n$

(B) $T(n) \leq T(n/5) + T(4n/5) + n$

(C) $T(n) \leq 2T(4n/5) + n$

(D) $T(n) \leq 2T(n/2) + n$

SOLUTION

Pivot element is found in Quick sort in every iteration all the elements to its left are smaller than & all in the right are greater than it.
So if $1/5^{th}$ of sorted sequence is the pivot. So sequence is divided into $1/5^{th}$ & $4/5^{th}$ of the sequence.
So recursion will be

$$T(n) = T(n/5) + T\left(\frac{4n}{5}\right) + n$$

Hence (B) is correct option.

Question. 87

The subset-sum problem is defined as follows: Given a set S of n positive integers and a positive integer W; determine whether there is aa subset of Swhose elements sum to W.

An algorithm Q Solves this problem in $O(nW)$ time. Which of the following statements is false?

(A) Q sloves the subset-sum problem unpolynomial time when the input is encoded in unary

(B) Q solves the subset-sum problem is polynominal time when the input is encoded in binary

(C) The subset sum problem belongs to the class NP

(D) The subset sum problem in NP-hard

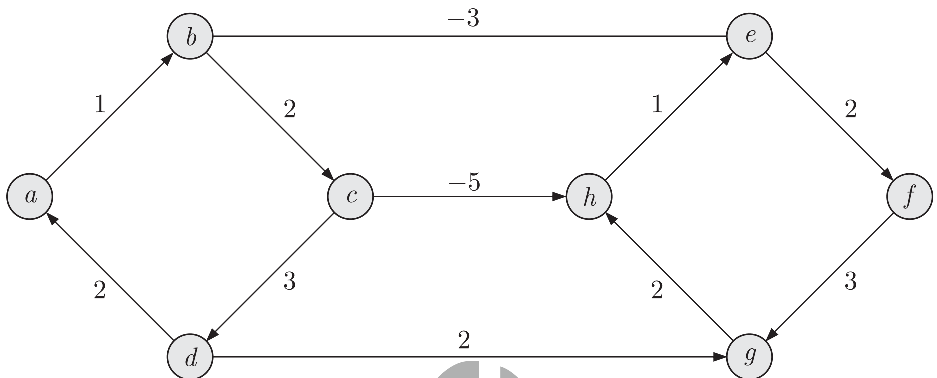
SOLUTION

W is an integer so the time taken by the algorithm is $\Theta(n)$ only. Since subset problem is NP complete, it should be in class NP & NP hard, so option (C) & (D) are true.
Using unary integer the algorithm will be solved in $\Theta(n)$ time but using binary it would take more time depending upon no. of bits.
So option (B) is false

Hence (B) is correct option.

Question. 88

Dijkstra's single source shortest path algorithm when run from vertex a in the above graph, computes the corrects shortest path distance to



(A) only vertex a

(B) only vertices a, e, f, g, h

(C) only vertices, a, b, c, d

(D) all the vertices

SOLUTION

We apply Dijkstra Algorithm.

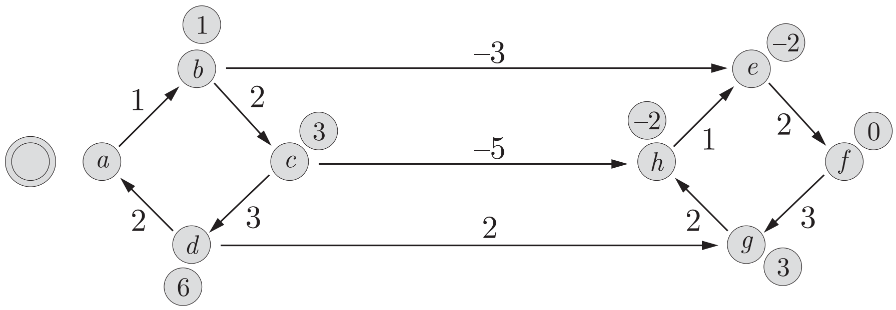
Vertex	I(a)	II(a)	III(b)	IV(e)	V(f)	VI(c)	VII(h)	VIII(g)
I	a	0	0	0	0	0	0	0
I	b	$\underline{1}$	1	1	1	1	1	1
V	c	∞	$\underline{3}$	3	3	3	3	3
III	d	∞	∞	∞	∞	$\underline{6}$	6	6
III	e	∞	$\underline{-2}$	-2	-2	-2	-2	-2
IV	f	∞	∞	$\underline{0}$	0	0	0	0
VII	g	∞	∞	∞	$\underline{3}$	3	3	3
VI	h	∞	∞	∞	-2	$\underline{\underline{-2}}$	-2	-2

Order $a\,b\,e\,f\,c\,h\,g\,d$

$a \to b = 1 \quad a \to c = 3 \quad a \to b = 6$

$a \to e = -2 \quad a \to f = 0 \quad a \to g = 3$

$a \to h = -2$



Since there is no $-ve$ cycle so Dijkstra gives correct result for all vertices.
Hence (D) is correct option.

Question. 89

You are given the postorder traversal, P of $a$binary search tree on the n elements 1,2,.....,n. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

(A) $\Theta(\log n)$

(B) $\Theta(n)$

(C) $\Theta(n \log n)$

(D) none of the above, as the tree cannot be uniquely determined.

SOLUTION

To construct a BST from post order we also require in-order traversal since given the elements are 1 2.......n So their sorted order would be in order.
Using both BST can be constructed in a linear scan. So it will take only Θn time.
Hence (B) is correct option.

Question. 90

We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is

(A) $\Theta(\log n)$ (B) $\Theta(n)$

(C) $\Theta(n \log n)$ (D) $\Theta(n^2)$

SOLUTION

Heaps are implemented as simple arrays, to insert n more elements each element take $\Theta(1)$ time.
So total time would be $\Theta(n)$.
Hence (D) is correct option.

Common data for Questions 91 & 92:

Consider the following C functions:

```
int f1 (int n)
{
      If(n==0||n==1)
            return n;
      else
            return(2*f1(n-1)+3*f1(n-2));
}
int f2(int n)
{
      int i;
      int X[N], Y[N], Z[N];
      X[1]=1;Y[1]=2;Z[1]=3;
      for  (i=2;i<=n;i++){
           X[i]=Y[i-1]+Z[i-2];
           Y[i]=2*X[i];
           z[i]=3*X[i];
      }
      return X[n];
}
```

Question. 91

The running time of $f1(n)$ and $f2(n)$ are

(A) $\Theta(n)$ and $\Theta(n)$
(B) $\Theta(2^n)$ and $O(n)$
(C) $\Theta(n)$ and $\Theta(2^n)$
(D) $\Theta(2^n)$ and $\Theta(2^n)$

SOLUTION

Procedure f_1 has the nature

$$T(0) = 0,\ T(1) = 1 = 2^0$$
$$T(n) = 2T(n-1) + 3T(n-2)$$

Solution to this recursion is 2^n
So $\Theta(2^n)$
Procedure f_2 has a single for loop from 2 to n so the complexity will be $\Theta(n)$.
Hence (B) is correct option.

Question. 92

$f1$ (8) $f2$ (8) return the values

(A) 1661 and 1640
(B) 59 and 59
(C) 1640 and 1640
(D) 1640 and 1661

SOLUTION

$f_1(8)$

n	Value	Return
8	$2f1(7)+3f1(6)$	$1094+546=1640$
7	$2f1(6)+3f1(5)$	$364+183=547$
6	$2f1(5)+3f1(4)$	$122+60=182$
5	$2f1(4)+3f1(3)$	$40+21=61$
4	$2f1(3)+3f1(2)$	$14+6=20$
3	$2f1(2)+3f1(1)$	$4+3=7$
2	$2f1(1)+3f1(0)$	2
1	1	1
0	0	0

$$f(1)(8)=1640$$

Now

$$X[1]=1$$
$$Y[1]=2$$
$$Z[1]=3$$

Iteration	X[i]	Y[i]	Z[i]
2	$2+0=2$	4	6
3	$4+3=7$	14	21
4	$14+6=20$	40	60
5	$40+21=61$	122	183
6	$122+60=182$	364	546

7	$364+183=547$	1094	1641
8	$1094+546=1640$		

Return $X[8]=1640$ finally

Hence (C) is correct option.

Statement for Linked Answers Questions 93 & 94:

The subset-sum problem is defined as follows. Given set of n positive integers, $S=\{a_1, a_2, a_3, \ldots\ldots a_n\}$ and a positive integer W is there a subset S whose elements sum of W? A dynamic program for solving this problem uses a 2-dimensiond Boolean array, X with n rows and W-1 columns $X[i,j], 1 \le i \le W$, is TRUE if and only if there is a subset of $\{a_1, a_2, \ldots\ldots a_i\}$ whose elements sum to j.

Question. 93

Which of the following is valid for $2 \le i \le n$ and $a_1 \le j \le W$?

(A) $X[i,j] = X[i-1,j] \vee X[i,j-a_i]$

(B) $X[i,j] = X[i-1,j] \vee X[i-1,j-a_i]$

(C) $X[i,j] = X[i-1,j] \wedge X[i,j-a_i]$

(D) $X[i,j] = X[i-1,j] \wedge X[i-1,j-a_i]$

SOLUTION

Dynamic programming can be successfully used, i.e n rows for n elements & $w+1$ columns.

Each row is filled on the basis of its previous rows & the $(j-a_i)^{th}$ column.

If any of them is 0 then $X[i,j]$ should be zero.

This require $X[i,j] = X[i-1,j]\ V \times [i-1,j-a_i]$

Hence (B) is correct option.

Question. 94

Which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?

(A) $X[1,W]$ (B) $X[n,0]$

(C) $X[n,W]$ (D) $X[n-1,n]$

SOLUTION

The algorithm of dynamic programming has n rows & w columns. These would be filled dynamically depending upon previous rows & columns. So $X[n,w]$ will be filled in the last & this would give the result.
If $X[n,w]=1$ i.e. TRUE, then we know that there is subset present whose sum $=$ integer w.
Otherwise subset not present.
Hence (C) is correct option.

Statement for Linked Answers Questions 95 & 96:

Consider the following C program that attempts to locate an element x in an array $Y[\,]$ using binary search. The program is erroneous.

```
          1
        do {
                2
        if        i
    } while             i
    if           print f   is in the array");
        else printf    is not in the array");
    }
```

Question. 95

On which of the following contents of Y and x does the program fail?

(A) Y is [1 2 3 4 5 6 7 8 9 10] and $x<10$

(B) Y is [1 3 5 7 9 11 13 15 17 19] and $x<1$

(C) Y is [2 2 2 2 2 2 2 2 2 2] and $x>2$

(D) Y is [2 4 6 8 10 12 14 16 18 20] and $2<x<20$ and x is even

SOLUTION

Running on option (C)
$i=0$ to 9 X let 3 2222222222

i	j	K	$y[K]$	$y[K]!=X$	$i<j$
0	$9\rightarrow 5$	5	2	True	True

0	$5 \to 3$	3	2	True	True
0	$3 \to 2$	2	2	True	True
0	$2 \to 1$	1	2	True	True
0	1	1	2	True	True
0	1	1	2	True	True

It would never stop & we are looking for no. X >2 but we are moving in wrong dir.
Hence (C) is correct option.

Question. 96

The correction needed in the program to make it work properly is

(A) change line 6 to : if $(Y[k]) < x)\, i = k+1;$else j = k - 1;

(B) change line 6 to: if $(Y[k] < x)\, i = k-1;$else j = k + 1;

(C) change line 6 to: if $(Y[k] < x)\, i = k;$ else $j = k;$

(D) change line 7 to : } while $((Y[k] == x) \&\& (i < j));$

SOLUTION

A slight change can be made which will prevent this loop to go infinite in

line 6 : if $Y(K) < x$
$i = K+1;$
else
$j = K-1;$

Should be there.
Hence (A) is correct option.

YEAR 2009

Question. 97

What is the number of swaps required to sort n elements using selection sort, in the worst case ?

(A) $\theta(n)$ (B) $\theta(n\log n)$

(C) $\theta(n^2)$ (D) $\theta(n^2\log n)$

SOLUTION

In selection sort the worst case would be when the elements are reverse sorted, here the algorithm selects the min element the first element, and during linear scan if element found min then a swap takes place.

So during n iterations maximum.

n swap can occur in each iteration.

$$\text{No. of swaps} = n[n + (n-1) + (n-2)...........1]$$
$$= \Theta(n^2)$$

Hence (C) is correct option.

Question. 98

Which of the following statement(s) is/are correct regarding Bellman-Ford shortest path algorithm ?

P. Always finds a negative weighted cycle, if one exists.

Q. Finds whether any negative weighted cycle is reachable from the source

(A) P only (B) Q only

(C) Both P and Q (D) Neither P nor Q

SOLUTION

Bellman ford Algorithm is used when there are negative weights assigned to the edges, this can cause generation of $-ve$ cycles, reached from the source vertex. So both statements are correct.

Hence (C) is correct option.

Question. 99

Let π_A be a problem that belongs to the class NP. Then which one of the following is TRUE ?

(A) There is no polynomial time algorithm for π_A

(B) If π_A can be solved deterministically in polynomial time, then $P = NP$

(C) If π_A is NP-hard, then it is NP-complete

(D) π_A may be undecidable

SOLUTION

Problems which are both NP & NP hard are called NP complete problems. So option (C) is correct option.

(A) Can't be correct since $P \in NP$, so there can be any Algorithm with P time.

(B) Is not true, if some solved deterministically but if not NP complete then can't be $P = NP$.

(D) Is not correct because some problems which are NP decidable under certain conditions.

Question. 100

The running time of an algorithm is represented by the following recurrence relation:

$$T(n) = \begin{cases} n & n \leq 3 \\ T\left(\frac{n}{3}\right) + cn & \text{otherwise} \end{cases}$$

Which one of the following represents the time complexity of the algorithm ?

(A) $\theta(n)$ (B) $\theta(n \log n)$

(C) $\theta(n^2)$ (D) $\theta(n^2 \log n)$

SOLUTION

For $n \leq 3$ $T(n) = n$

i.e $\Theta(n)$

For $n > 3$

$$T(n) = T(n/3) + Cn$$

This can be solved by Master's Theorem

$$a = 1 \;\; b = 2 \quad \log_b a = \log_3 1 = 0$$

$$f(n) = Cn$$

$$n^{\log_b a} < f(n)$$

$$n^{\circ} < f(n)$$

So, $T(n) = \theta(f(n))$ case III

$$= \theta(Cn)$$

$$= \theta(n)$$

Whole complexity $\forall n$

$$T(n) = \theta(n) + \theta(n)$$

$$= \theta(n)$$

Hence (A) is correct option.

Question. 101

The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \bmod 10$ and linear probing. What is the resultant hash table ?

(A)

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(C)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(D)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	
6	
7	
8	
9	

SOLUTION

$$K = 12, 18, 13, 2, 3, 23, 5, 15$$
$$h \;\; K \bmod 10$$

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

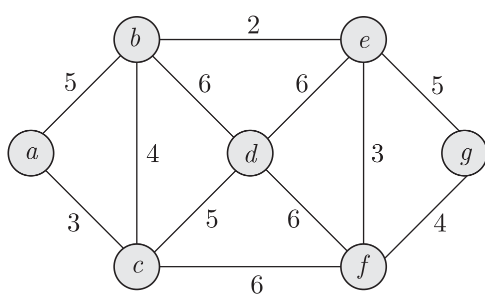
Step 1
3
4
5
6
7

Step 2
8

Hence (C) is correct option.

Question. 102

Consider the following graph:



Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm ?

(A) (b, e) (e, f) (a, c) (b, c) (f, g) (c, d)

(B) (b, e) (e, f) (a, c) (f, g) (b, c) (c, d)

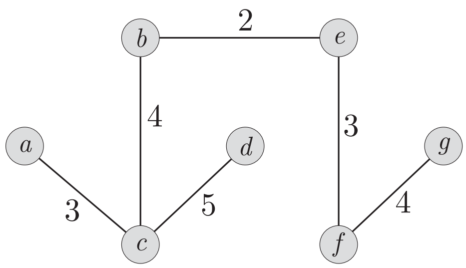
(C) (b, e) (a, c) (e, f) (b, c) (f, g) (c, d)

(D) (b, e) (e, f) (b, c) (a, c) (f, g) (c, d)

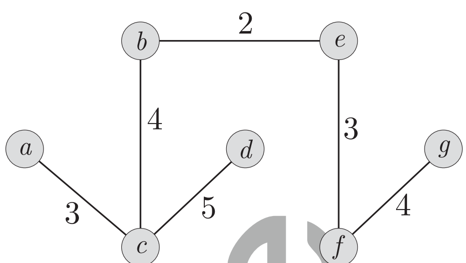
SOLUTION

Krushal's algorithm, arranging edges in ascending order.

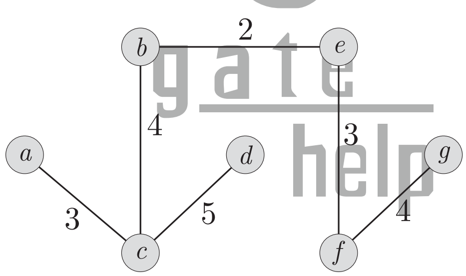
{2, 3, 3, 4, 4, 5, 5, 6, 6, 6, 6}



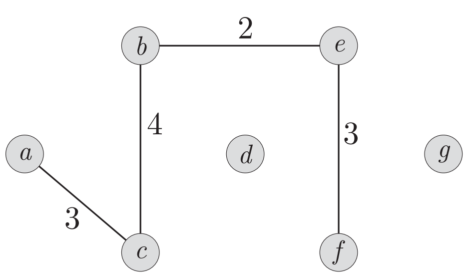
(A)



(B)



(C)



(D)

(A, C) can't be taken after (B, C).
Hence (D) is correct option.

Question. 103

In quick sort, for sorting n elements, the $(n/4^{th})$ smallest element is selected as pivot using an $O(n)$ time algorithm. What is the worst case time complexity of the quick sort ?

(A) $\theta(n)$
(B) $\theta(n\log n)$
(C) $\theta(n^2)$
(D) $\theta(n^2\log n)$

SOLUTION

Pivot in guide sort is the index which is sorted in that run, all the elements in its left are smaller than it & elements greater than it are on its right side.

So the recursion becomes.

$$T(n) = T(n/4) + T(3n/4) + n$$

Solution to this recursion is $\theta(n\log n)$

Hence (B) is correct option.

Common Data for Question 104 & 105 :

A sub-sequence of a given sequence is just the given sequence with some elements (possibly none of all) left out. We are given two sequence $X[m]$ and $Y[n]$ of length m and n, respectively, with indexes of X and Y starting from 0.

Question. 104

We wish to find the length of the longest common sub-sequence (LCS) of $x[m]$ and $Y[n]$ of lengths m and n, where an incomplete recursive definition for the function $l(i,j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below :

$l(i,j)$ =0, if either $i=0$ or $j=0$

=expr1, if $i,\ j>0$ and $x[i-1] = Y[j-1]$

=expr2, if $i,\ j>0$ and $x[i-1] \neq Y[j-1]$

Which one of the following options is correct ?

(A) expr 1$\equiv$ $l(i-1,j)+1$

(B) expr 1$\equiv$ $l(i,j-1)$

(C) expr 2$\equiv$ $\max(l(i-1,j), l(i,j-1))$

(D) expr 2$\equiv$ $\max(l(i-1,j-1), l(i,j))$

SOLUTION

LCS problem is solved using dynamic programming in which every row is dependent upon previous rows & columns. If two literals at i^{th} row & j^{th} column doesn't match then.

Eve fill $l[i, j]$ with max $[L[i-1,j],[i,j-1]]$ i.e max of previous cell in row & column.

So $$\text{expr } 1 = L[i-1, j-1] + 1$$
$$\text{expr } 2 = \max([l[i-1,j], L[i,j-1]])$$

Hence (C) is correct option.

Question. 105

The values of $l(i,j)$ could be obtained by dynamic programming based on the correct recursive definition of $l(i,j)$ of the form given above, using an array $L[M,N]$, where $M = m+1 \text{and} N = n+1$, such that $L[i,j] = l(i,j)$.

Which one of the following statements would be true regarding the dynamic programming solution for the recursive definition of $l(i,j)$?

(A) All elements of L should be initialized to 0 for the values of $l(i,j)$ to be properly computed.

(B) The values of $l(i,j)$ may be computed in a row major order or column major order of $L[M,N]$

(C) The values of $l(i,j)$ cannot be computed in either row major order or column major order of $L[M,N]$

(D) $L[p,q]$ needs to be computed before $L[r,s]$ if either $p < r$ or $q < s$

SOLUTION

During solution through dynamic programming option (B) & (C) are incorrect since the solution is done in row major order. But not in column major order.

(A) is true but not necessary.

(D) is correct eg.

$L[4,5]$ require

$L[3,4]\,L[3,5]$ here (3, 4), (3, 5) & (4, 4)

$L[4,4]\,L[4,5]$ need to be calculated before [4,5]

So $L[p,q]$ required to be calculated before $L[r,s]$ if $p < r$ or $q < s$.

Hence (D) is correct option.

YEAR 2010

Question. 106

Two alternative package A and B are available for processing a database having 10^k records. Package A requires 0.0001 n^2 time units and package B requires $10n\log_{10}n$ time units to process n records. What is the smallest value of k for which package B will be preferred over A ?

(A) 12 (B) 10

(C) 6 (D) 5

SOLUTION

$$\text{No. of record} = 10^K$$

$$K = ?\ \ A = .0001n^2 \qquad B = 10n\log_{10}^{n}$$

$$A = \frac{n^2}{10^5}$$

Required time A $>$ time B.

$$\frac{n^2}{10^5} > 10\,n\log_{10}n$$

So here the co.eff. is 10^6

So smallest value of $K = 6$

Hence (C) is correct option.

Question. 107

The weight of a sequence $a_0, a_1 \ldots\ldots, a_{n-1}$ of real numbers is defined as $a_0 + a_1/2 + \ldots\ldots a_{n-1}/2^{n-1}$. A subsequence of a sequence is obtained by deleting some elements form the sequence, keeping the order of the remaining elements the same. Let X denote the maximum possible weight of a subsequence of $a_0, a_1, \ldots\ldots a_{n-1}$ and Y the maximum possible weight of a subsequence of $a_1, a_2, \ldots\ldots, a_{n-1}$. Then X is equal to

(A) max $(Y, a_0 + Y)$ (B) max $(Y, a_0 + Y/2)$

(C) max $(Y, a_0 + 2Y)$ (D) $a_0 + Y/2$

SOLUTION

$$\text{Given X } a_0 + \frac{a_1}{2} + \frac{a_2}{2^2} + \ldots\ldots \frac{a_{n-1}}{2^{n-1}} \qquad (1)$$

$$Y = a_1 + \frac{a_2}{2} + \frac{a_3}{2^2} + \ldots\ldots\ldots\ldots \frac{a_{n-1}}{2^{n-2}} \qquad (2)$$

Dividing eq. (2) by 2

$$\frac{Y}{2}\ \frac{a_1}{2} + \frac{a_2}{2^2} + \frac{a_3}{2^3} \ldots\ldots\ldots\ldots \frac{a_{n-1}}{2^{n-1}} \qquad (3)$$

Putting eq. (3) in eq. (1)

$$X = a_0 + \frac{Y}{2}$$

Hence (D) is correct option.

Common Data for Questions 108 & 109

Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry W_{ij} in the matrix W below is the weight of the edge $\{i, j\}$.

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

Question. 108

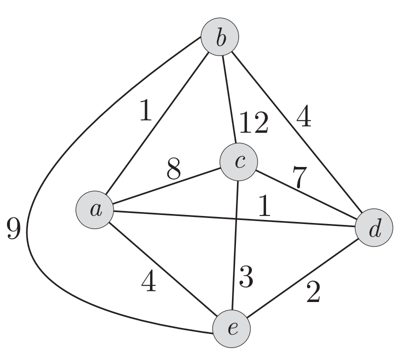
What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T ?

(A) 7 (B) 8

(C) 9 (D) 10

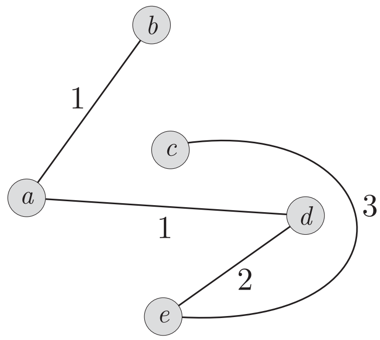
SOLUTION

let $\{0, 1, 2, 3, 4\}$ be $\{a, b, c, d, e\}$



Drawing spanning true using Prim's Algorithms start with a

(A)	$\Rightarrow (a, b), (a, c), (a, d), (a, e)$	$(a, b) = 1$
(B)	$\Rightarrow (a, c), (a, d), (a, e), (b, c), (b, d), (b, e)$	$(a, d) = 1$
(D)	$\Rightarrow (a, c), (a, e), (b, c), (b, d), (b, e), (c, d)$	$(d, e) = 2$
(E)	$\Rightarrow (a, c), (a, e), (b, c), (b, d), (b, e), (c, d), (c, e)$	$(c, e) = 3$



$$\text{Weight} = 1 + 1 + 2 + 3$$
$$= 7$$

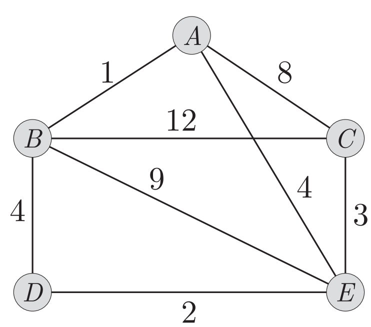
Question. 109

What is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges ?

(A) 7
(B) 8
(C) 9
(D) 10

SOLUTION

Min possible path from B to C so we draw all paths from B to C



$$B \to C = 12$$
$$B \to A \to C = 1 + 8 = 9$$
$$B \to E \to C = 9 + 3 = 12$$

$$B \to \triangle \to E \to C = 4 + 2 + 3 = 9$$
$$B \to A \to E \to C = 1 + 4 + 3 = 8$$

This is min path.

Hence (B) is correct option.

****

GATE Multiple Choice Questions
For Computer Science

By NODIA and Company

Available in Two Volumes

FEATURES

- The book is categorized into units and the units are sub-divided into chapters.
- Chapter organization for each unit is very constructive and covers the complete syllabus
- Each chapter contains an average of 40 questions
- The questions are standardized to the level of GATE examination
- Solutions are well-explained, tricky and consume less time. Solutions are presented in such a way that it enhances you fundamentals and problem solving skills
- There are a variety of problems on each topic
- Engineering Mathematics is also included in the book